

Character Strings

As in Fortran 77, all characters have a fixed length, but there are some new intrinsic functions for the manipulation of these strings.

trim returns a string with all trailing blanks removed. len_trim (string) is the length of trim (string); it is the position of the last nonblank character in the string.

```
character (len = 3) :: string_a = "a" ,   &
                      string_b = "bb"
character (len = 6) :: string_c
.
.
.
string_c = string_a // string_b ! = "a  bb "
string_c = trim (string_a) // string_b
      ! = "abb  "
```

[Next slide](#)

[Learn more about the character type.](#)

Substrings and Concatenation

Concatenating strings (as shown in the example on the previous slide) and accessing substrings are two common operations performed on character strings.

```
name = "John X. Public"  
name(6:7) = "Q."
```

[Learn more about substrings](#). [Previous slide](#) [Next slide](#)

index (string, substring) finds the first position in string where substring occurs, or 0 if it does not occur. There is an optional argument back, which indicates that the search should start at the right end.

```
index ("monkey", "on") ! = 2
index ("monkey", "off") ! = 0
index ("banana", "ana") ! = 2
index ("banana", "ana", back=.true.) ! = 4
```

[Previous slide](#) [Next slide](#)

scan

The function `scan (string, characters)` returns the first position in `string` occupied by any one of the characters in `characters`; it returns 0 if none of the characters occurs in `string`. `scan` also has an optional `back` argument.

```
scan ("abcabcabc", "bc") ! = 2
scan ("abcabcabc", "bc", .true.) ! = 9
scan ("abcabcabc", "defg") ! = 0
scan ("abcabcabc", "cab") ! = 1
```

[Previous slide](#) [Next slide](#)

verify

The function `verify (string, characters)` returns the first position in `string` occupied by any character *not* in `characters`; it returns 0 if all of the characters in `string` occur in `characters`. `verify` also has an optional `back` argument.

```
verify ("abcababcabc", "bc") ! = 1  
verify ("abcababcabc", "bc", .true.) ! = 7  
verify ("abcababcabc", "defg") ! = 1  
verify ("abcababcabc", "cab") ! = 0
```

[Learn more about character intrinsic functions.](#) [Previous slide](#) [Next slide](#)

The user-defined function `int_char` uses `index` and `verify` to convert a string of decimal digits into an integer value. The result is -1 if there is a character other than a digit in the string. The result is -2 if the number is too large to represent as an integer. [Previous slide](#) [Next slide](#)

intchar2

```
function int_char (string)  &
    result (int_char_result)
! Converts digit string to an integer
! If nondigit in string, value is -1
! If number is too large, value is -2

    integer :: int_char_result
    character (len = *), intent (in) :: string
    character (len = *), parameter ::  &
        dec_digits = "0123456789"
    integer, parameter ::  &
        max_result = (huge (0) - 9) / 10
    integer :: digit
```

[Previous slide](#) [Next slide](#)

```
intchar3
if (verify (string, dec_digits) /= 0) then
    int_char_result = -1
    return
end if
int_char_result = 0
do digit = 1, len (string)
    if (int_char_result > max_result) then
        int_char_result = -2
        return
    else
        int_char_result = int_char_result * 10 + &
index (dec_digits, string (digit:digit)) - 1
    end if
end do
end function int_char
```

[Previous slide](#) [Next slide](#)

Exercise

1. Write a logical valued function `Fortran_name` that determines whether or not its character string argument is legal Fortran name.

[Previous slide](#)